

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

---

In re Letters Patent of:  
Rabin Sugumar et al.

Patent No.: 7,340,590

Issued: March 4, 2008

For: HANDLING REGISTER DEPENDENCIES  
BETWEEN INSTRUCTIONS SPECIFYING  
DIFFERENT WIDTH REGISTERS

---

**REQUEST FOR CERTIFICATE OF CORRECTION  
PURSUANT TO 37 CFR 1.323 AND 1.322**

Attention: Certificate of Correction Branch  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

Upon reviewing the above-identified patent, Patentee noted typographical errors which should be corrected. A listing of the errors to be corrected is attached.

The typographical errors marked with an "A" on the attached list are found in the application as filed by applicant. Payment in the amount of \$100.00 covering the fee set forth in 1.20(a) is enclosed.

The typographical errors marked with a "P" on the attached list are not in the application as filed by applicant. Also given on the attached list are the documents from the file history of the subject patent where the correct data can be found.

The errors now sought to be corrected are inadvertent typographical errors the correction of which does not involve new matter or require reexamination.

Transmitted herewith is a proposed Certificate of Correction effecting such corrections.  
Patentee respectfully solicits the granting of the requested Certificate of Correction.

The Commissioner is authorized to charge any deficiency of up to \$300.00 or credit any excess in this fee to Deposit Account No. 04-0100.

Dated: March 26, 2008

Respectfully submitted,

By  \_\_\_\_\_

Flynn Barrison

Registration No.: 53,970

DARBY & DARBY P.C.

P.O. Box 770

Church Street Station

New York, New York 10008-0770

(212) 527-7700

(212) 527-7701 (Fax)

Attorneys/Agents For Applicant

## UNITED STATES PATENT AND TRADEMARK OFFICE CERTIFICATE OF CORRECTION

PATENT NO. : 7,340,590

Page 1 of 1

APPLICATION NO.: 10/734,763

ISSUE DATE : 10/734,763

INVENTOR(S) : Sugumar et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the face page, in filed (56), under "Other Publications", in column 2, line 3, delete "naming.as" and insert - - naming as - -, therefor.

On Sheet 1 of 5, in Fig. 1(Box 110), line 1, delete "SUBSYSTEM" and insert - - SUBSYSTEM - -, therefor.

In column 10, line 59, in Claim 4, delete "in stead" and insert - - instead - -, therefor.

In column 11, line 19, in Claim 8, delete "instruction" and insert - - instruction, - -, therefor.

In column 11, line 41, in Claim 11, delete "modified" and insert - - modifies - -, therefor.

In column 12, line 66, in Claim 25, delete "modifying," and insert - - modifying - -, therefor.

In column 13, line 3, in Claim 25, delete "produce" and insert - - producer - -, therefor.

In column 13, line 7, in Claim 25, delete "find" and insert - - and - -, therefor.

In column 14, line 25, in Claim 30, delete "dependence" and insert - - dependency - -, therefor.

### MAILING ADDRESS OF SENDER (Please do not use customer number below):

John W. Branch, Esq.  
DARBY & DARBY P.C.  
P.O. Box 770  
Church Street Station  
New York, New York 10008-0770

1

This collection of information is required by 37 CFR 1.322, 1.323, and 1.324. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1.0 hour to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Attention Certificate of Corrections Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

# Darby & Darby

Issued Patent Proofing Form

File#: 20910/0205488-US0

Note: P = PTO Error

A = Applicant Error

Proofread By: Shabana (03/17/2008)

US Serial No.: 10/734,763

US Patent No.: US 7,340,590 B1

Issue Dt.: Mar. 4, 2008

Title: HANDLING REGISTER DEPENDENCIES BETWEEN INSTRUCTIONS SPECIFYING DIFFERENT WIDTH

## REGISTERS

Proofing Instructions: Whole Patent

Sr. No.	P/A	Original		Issued Patent		Description of Error
		Page	Line	Column	Line	
1	A	Page 1 of 1 List of References cited by applicant and considered by examiner (03/14/2006)	Entry 1 Line 3 (Other Art)	First Page Col. 2 (Other Publication)	3	Delete "naming.as" and insert - - naming as - -, therefor.
2	A	Sheet 1 of 5 Drawings-only black and white line drawings (12/11/2003)	1 (Box 110) (FIG. 1)	Sheet 1 of 5 (Box 110) (FIG. 1)	1	Delete "SUBSYSTEM" and insert - - SUBSYSTEM - -, therefor.
3	P	Page 3 Claims (08/17/2007)	Claim 6 Line 5	10	59	In Claim 4, delete "in stead" and insert - - instead - -, therefor.
4	P	Page 3 Claims (08/17/2007)	Claim 10 Line 7	11	19	In Claim 8, delete "instruction" and insert - - instruction, - -, therefor.
5	P	Page 4 Claims (08/17/2007)	Claim 13 Line 2	11	41	In Claim 11, delete "modified" and insert - - modifies - -, therefor.
6	P	Page 7 Claims (08/17/2007)	Claim 31 Line 4	12	66	In Claim 25, delete "modifying," and insert - - modifying - -, therefor.
7	P	Page 7 Claims (08/17/2007)	Claim 31 Line 6	13	3	In Claim 25, delete "produce" and insert - - producer - -, therefor.
8	P	Page 7 Claims (08/17/2007)	Claim 31 Line 9	13	7	In Claim 25, delete "find" and insert - - and - -, therefor.
9	P	Page 8 Claims (08/17/2007)	Claim 36 Line 6	14	25 (Approx.)	In Claim 30, delete "dependence" and insert - - dependency - -, therefor.



US007340590B1

(12) **United States Patent**  
**Sugumar et al.**

(10) Patent No.: **US 7,340,590 B1**  
 (45) Date of Patent: **Mar. 4, 2008**

- (54) **HANDLING REGISTER DEPENDENCIES BETWEEN INSTRUCTIONS SPECIFYING DIFFERENT WIDTH REGISTERS**
- (75) Inventors: **Rabin Sugumar, Sunnyvale, CA (US); Sorin Iacobovici, San Jose, CA (US); Chandra M. R. Thimmannagari, Fremont, CA (US)**
- (73) Assignee: **Sun Microsystems, Inc., Santa Clara, CA (US)**
- (\*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 357 days.
- (21) Appl. No.: **10/734,763**
- (22) Filed: **Dec. 11, 2003**
- (51) Int. Cl. **G06F 9/312** (2006.01)
- (52) U.S. Cl. **712/216; 712/210**
- (58) Field of Classification Search **712/216, 712/210, 211; 708/550**
- See application file for complete search history.
- (56) **References Cited**

**U.S. PATENT DOCUMENTS**

4,679,140 A *	7/1987	Gotou et al.	712/229
4,847,802 A *	7/1989	Ashton	708/513
5,446,912 A *	8/1995	Colwell et al.	712/217
5,515,520 A *	5/1996	Hatta et al.	708/550
5,630,149 A *	5/1997	Bluhm	712/23
5,678,016 A *	10/1997	Eisen et al.	712/216
5,732,005 A *	3/1998	Kahle et al.	708/495
5,778,247 A *	7/1998	Tremblay	712/23
5,790,827 A *	8/1998	Leung	712/216
5,805,475 A *	9/1998	Putrino et al.	708/204
6,029,243 A *	2/2000	Pontius et al.	712/222
6,094,719 A *	7/2000	Panwar	712/216
6,105,129 A *	8/2000	Meier et al.	712/222
6,195,746 B1 *	2/2001	Nair	712/217

6,216,200 B1 *	4/2001	Yeager	711/100
6,237,076 B1 *	5/2001	Gaertner et al.	712/23
6,253,312 B1 *	6/2001	Elliott et al.	712/222
6,442,676 B1 *	8/2002	Guenther	712/210
6,449,710 B1 *	9/2002	Isaman	712/216
6,463,525 B1 *	10/2002	Prabhu	712/222
6,493,819 B1 *	12/2002	Mahurin et al.	712/210
7,124,286 B2 *	10/2006	McGrath et al.	712/229
2002/0174319 A1 *	11/2002	Rivers et al.	712/200
2004/0148492 A1	7/2004	Sugumar et al.	

**OTHER PUBLICATIONS**

U.S. Appl. No. 10/728,039, titled "Method and Processor for Facilitating Greater-Width Store Operations of Lesser-Width Source Values" filed Dec. 4, 2003, namings as inventors Atula Kalambur et al.

\* cited by examiner

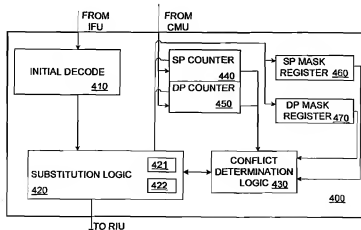
Primary Examiner—Eric Coleman

(74) Attorney, Agent, or Firm—Darby & Darby P.C.; M. David Ream

(57) **ABSTRACT**

The present application describes a method and a processor for handling register dependency conflicts between lesser and greater width instructions, colloquially referred to as "evil twins." If there is a register dependency between a greater width producer instruction and a lesser width consumer instruction, a greater width source register is substituted for the source register specified by the lesser width producer. If there is a register dependency between a lesser width producer instruction and a greater width producer instruction, the greater width consumer instruction is replaced by multiple helper instructions. One or more of the helper instructions merge lesser width registers aliased onto the source registers specified by the greater width consumer instruction, into temporary registers. Another helper instruction executes the greater width consumer instruction using the temporary registers instead of the original source registers.

**30 Claims, 5 Drawing Sheets**



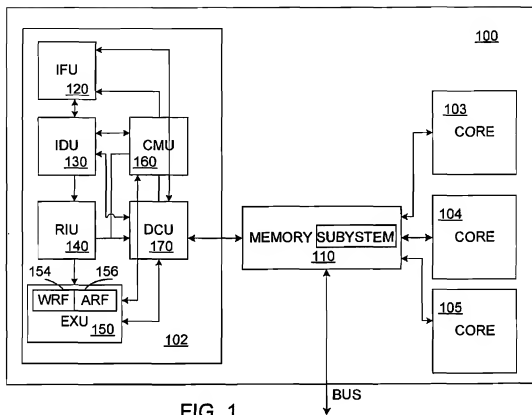


FIG. 1

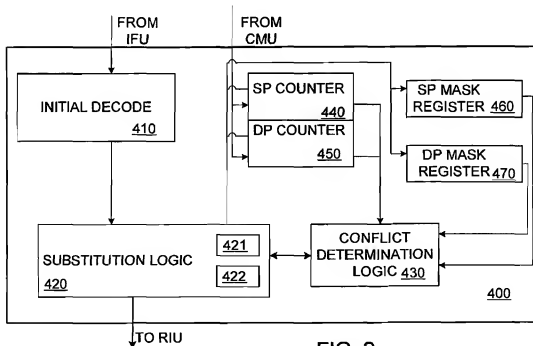


FIG. 2

the double precision instruction both modify a common register, and if an instruction being currently decoded specifies that common register as a source, then an SP producer/DP producer conflict exists, and the method proceeds to 540. At 540 the method generates an indication, e.g. asserts a signal, sets a control bit, or the like, that an SP producer/DP producer conflict exists. If no SP producer/DP producer conflict is identified, then the method proceeds to 545 to begin checking for an SP producer/DP consumer conflict.

At 545, the method checks whether there are any active SP modifying instructions in the pipe. In at least one embodiment SP counter 440 and DP counter 450 (FIG. 2) are used for this purpose. If there are one or more active SP modifying instructions in the pipe, the method proceeds to 550. At 550, the method determines if there is a double precision consumer instruction in the fetch group being decoded. If not, then there is no double precision consumer to cause an SP producer/DP consumer conflict, and so the method proceeds to 565. If there is a double precision consumer in the fetch group, however, the method proceeds to 555. At 555 the method compares the register being modified by the active SP instruction in the pipe with the source registers specified by the double precision consumer instruction. This comparison is performed, in at least one embodiment, using the mask registers 460 and 470 illustrated in FIG. 2. If the method determines at 555 that the destination register of an active SP modifying instruction modifies the same register specified as a source of the double precision consumer instruction, the method generates an indication that an SP producer/DP consumer conflict exists at 560. Otherwise, the method begins checking for a DP producer/SP consumer conflict at 565.

At 565, the method checks whether there are any DP modifying instructions active in the pipe. As in 545, in at least one embodiment SP counter 440 and DP counter 450 (FIG. 2) are used for this purpose. If there are one or more active DP modifying instructions in the pipe, the method proceeds to 570. If not, then the method generates an indication that no conflict exists at 590.

At 570, the method determines if there is a single precision consumer instruction in the fetch group being decoded. If not, then there is no single precision consumer to cause a DP producer/SP consumer conflict, and so the method proceeds to 590. If there is a single precision consumer in the fetch group, however, the method proceeds to 575. At 575 the method compares the register being modified by the active DP instruction in the pipe with the source registers specified by the single precision consumer instruction. This comparison is performed, in at least one embodiment, using the mask registers 460 and 470 illustrated in FIG. 2. If the method determines at 575 that the destination register of an active DP modifying instruction modifies the same register specified as a source of the single precision consumer instruction, the method generates an indication that a DP producer/SP consumer conflict exists at 580. Otherwise, the method continues on to 590.

Those skilled in the art will appreciate that various techniques for comparing the registers specified as sources and/or destinations of instructions can be employed consistent with the teachings set forth herein. In addition, any number of signal generation techniques and/or hardware can be used to generate the signals discussed. It will also be appreciated that the order of action specified in methods 300 (FIG. 3) and 500 (FIG. 5) is flexible, and the various implementations may perform these actions in a different order than that illustrated.

Realizations in accordance with the present invention have been described in the context of particular embodiments. These embodiments are meant to be illustrative only and are not intended to be limiting. Many variations, modifications, additions, and improvements are possible and will become apparent to those skilled in the art upon consideration of the teachings set forth herein. Accordingly, plural instances may be provided for components described herein as a single instance. Boundaries between various components, operations and data stores are somewhat arbitrary, and particular operations are illustrated in the context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within the scope of claims that follow. Finally, structures and functionality presented as discrete components in the exemplary configurations may be implemented as a combined structure or component. These and other variations, modifications, additions, and improvements may fall within the scope of the invention as defined in the claims that follow.

What is claimed is:

1. A method comprising:

if a dependency exists between a greater width producer instruction and a lesser width consumer instruction, a processor modifying the lesser width consumer instruction by substituting for execution a greater width source register specifier for a lesser width source register specified in the lesser width consumer instruction; the processor executing the greater width producer instruction and placing a result of the execution in the greater width source register; and the processor executing the lesser width consumer instruction using the greater width source register.

2. The method of claim 1 wherein the greater width source register substituted for the lesser width source register is the greater width register onto which the lesser width source register is aliased.

3. The method of claim 2, wherein substituting the greater width register includes setting an indication that the lesser width source register is to be replaced by the greater width register.

4. A method comprising:

determining, if a dependency exists between a greater width instruction and a lesser width instruction; if a dependency exists between a greater width producer instruction and a lesser width consumer instruction, a processor modifying the lesser width consumer instruction by substituting for execution a greater width source register specifier for a lesser width source register specified in the lesser width consumer instruction, executing the greater width producer instruction and placing an execution result into the greater width source register, and executing the lesser width consumer instruction using the greater width source register; and

if a dependency exists between a lesser width producer instruction and a greater width consumer instruction, the processor substituting plural instructions for the greater width consumer instruction and executing the plural instructions in stead of the greater width consumer instruction.

5. The method of claim 4 further including stalling at least one instruction of a fetch group if a dependency exists between an instruction in the fetch group and both an active lesser width producer instruction and an active greater width producer instruction, until at least one of the active lesser width producer instruction or the active greater width producer instruction is retired, and then resuming execution.

11

6. The method of claim 4 wherein the greater width source register substituted for the lesser width source register is the greater width register onto which the lesser width source register is aliased.

7. The method of claim 6, wherein substituting the greater width register includes setting an indication that the lesser width source register is to be replaced by the greater width register.

8. The method of claim 4, wherein the plural instructions substituted for the greater width consumer instruction include:

a first instruction to merge plural lesser width registers aliased onto a first greater width source register of the greater width consumer instruction, the plural lesser width registers to be merged into a first temporary register;

a second instruction to merge plural lesser width registers aliased onto a second greater width source register of the greater width consumer instruction, the plural lesser width registers to be merged into a second temporary register; and

a third instruction to execute the greater width consumer instruction using the first temporary register and the second temporary register as source registers.

9. The method of claim 4, wherein determining if a dependency exists includes:

generating a first register mask identifying registers to be modified by lesser width instructions active in a pipeline; and

generating a second register mask identifying registers to be modified by greater width instructions active in the pipeline.

10. The method of claim 9, wherein determining if a dependency exists includes:

comparing a lesser width register specifier of an instruction against the second register mask; and

comparing a greater width register specifier of an instruction against the first register mask.

11. The method of claim 4, wherein determining if a dependency exists includes determining if a greater width instruction in a fetch group modified a lesser width source register specified by a younger instruction in the same fetch group.

12. The method of claim 4, wherein determining if a dependency exists includes determining if a lesser width instruction in a fetch group modifies a greater width source register specified by a younger instruction in the same fetch group.

13. A method of handling a register conflict between a first instruction specifying a greater width destination register and a second instruction specifying a lesser width source register, the method comprising a processor executing the second instruction using the greater width destination register as a source instead of the lesser width source register.

14. The method of claim 13, wherein the register specified by the greater width source specifier is a greater width register onto which the register specified by the lesser width source register specifier is aliased.

15. The method of claim 13, wherein substituting the greater width register specifier includes setting an indication that the register specified by the lesser width source register specifier is to be replaced by the greater width register.

16. An instruction decode unit including

logic to modify a lesser width consumer instruction by substituting a greater width source register specifier for a lesser width source register specifier in the lesser width consumer instruction if a dependency exists

12

between a greater width producer instruction and the lesser width consumer instruction.

17. The instruction decode unit of claim 16 further including logic to prevent at least one instruction from being delivered for execution if a dependency exists between an instruction in the fetch group and both an active lesser width producer instruction and an active greater width producer instruction.

18. The instruction decode unit of claim 16, wherein the logic to substitute the greater width register specifier includes logic to set an indicator in the lesser width source register specifier indicating that the lesser width source register is to be replaced by the greater width register.

19. A processor comprising:

an instruction decode unit including:

logic to modify a lesser width consumer instruction by substituting a greater width source register specifier for a lesser width source register specifier in the lesser width consumer instruction if a dependency exists between a greater width producer instruction and the lesser width consumer instruction, and

logic to substitute plural instructions for a greater width consumer instruction if a dependency exists between a lesser width producer instruction and a greater width consumer instruction.

20. The processor of claim 19 further including logic to prevent at least one instruction from being delivered for execution if a dependency exists between an instruction in a fetch group and both an active lesser width producer instruction and an active greater width producer instruction, until at least one of the active lesser width producer instruction or the active greater width producer instruction is retired, and then resuming execution.

21. The processor of claim 19, wherein the logic to substitute the greater width register specifier includes logic to set a bit in the lesser width source register specifier indicating that the lesser width source register is to be replaced by the greater width register.

22. The processor of claim 19, wherein the logic to substitute plural instructions for the greater width consumer instruction includes:

logic to generate a first instruction to merge plural lesser width registers aliased onto a first greater width source register specified by the greater width consumer instruction, the plural lesser width registers being merged into a first temporary register;

logic to generate a second instruction to merge plural lesser width registers aliased onto a second greater width source register specified by the greater width consumer instruction, the plural lesser width registers being merged into a second temporary register; and

logic to designate the first temporary register and the second temporary register as source registers of the greater width consumer instruction.

23. The processor of claim 19, further including logic to determine if a greater width producer instruction that is part of a fetch group modifies a lesser width source register specified by a younger instruction in the same fetch group.

24. The processor of claim 19, further including logic to determine if a lesser width producer instruction that is part of a fetch group modifies a greater width source register specified by a younger instruction in the same fetch group.

25. A processor comprising

means for determining if a dependency exists between a greater width instruction and a lesser width instruction; means for modifying the lesser width instruction by substituting a greater width source register specifier for

13

a lesser width source register specifier in the lesser width instruction if a dependency exists between a greater width produce instruction and a lesser width consumer instruction; and

means for substituting plural instructions for the greater width consumer instruction if a dependency exists between a lesser width producer instruction find a greater width consumer instruction.

26. The processor of claim 25, further including means for stalling at least one instruction in a fetch group if a dependency exists between an instruction in the fetch group and both an active lesser width producer instruction and an active greater width producer instruction, until at least one of the active lesser width producer instruction or the active greater width producer instruction is retired, and then resuming execution.

27. The processor of claim 25, wherein the means for substituting plural instructions for the greater width consumer instruction include:

means for generating a first instruction to merge plural lesser width registers aliased onto a first greater width source register of the greater width consumer instruction, the plural lesser width registers being merged into a first temporary register;

means for generating a second instruction to merge plural lesser width registers aliased onto a second greater width source register of the greater width consumer instruction, the plural lesser width registers being merged into a second temporary register; and

14

means for generating a third instruction to execute the greater width consumer instruction using the first temporary register and the second temporary register as source registers.

28. The processor of claim 25, wherein the means for determining if a dependency exists includes means for determining if a greater width destination instruction in a fetch group modifies a lesser width source register specified by a younger instruction in the same fetch group.

29. The processor of claim 25, wherein the means for determining if a dependency exists includes means for determining if a lesser width instruction in a fetch group modifies a greater width source register specified by a younger instruction in the same fetch group.

30. The processor of claim 19, wherein the instruction decode unit also includes: plural counters to track a number of instructions active in a pipeline;

plural mask registers to hold vectors indicating particular registers to be modified by active instructions; and

logic to compare destination and source registers specified by the instructions against at least one of the plural mask registers to determine if a dependence exists between an instruction being decoded and an active instruction.

\* \* \* \* \*